

Article

Optimization of Production Scheduling for the Additive Manufacturing of Ship Models Using a Hybrid Method

Kyeongho Kim ¹, Soonjo Kwon ² and Minjoo Choi ^{1,*}

¹ Department of Naval Architecture and Ocean Systems Engineering, Korea Maritime and Ocean University, Busan 49112, Republic of Korea; kkh0903@g.kmou.ac.kr

² School of Mechanical System Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea; soonjo.kwon@kumoh.ac.kr

* Correspondence: minjoo.choi@g.kmou.ac.kr

Abstract: This paper introduces a hybrid optimization method that leverages either linear programming (LP) or a genetic algorithm (GA) based on the problem size to enhance the parallel additive manufacturing (AM) process for ship models. The LP ensures optimality but can experience exponential increases in the computation time as the problem size grows. To address this limitation, the GA is employed for larger problems, providing optimal solutions within reasonable quality and time constraints. The method optimizes the module allocation to AM machines and determines the build processing sequence for each machine, while also considering the availability of workers preparing for consecutive module production. Applied to a case study, the proposed method achieves a 14% reduction in the completion time compared to a heuristic method from a previous study. Furthermore, the method is validated by benchmarking against the heuristic method across various problem sizes, consistently demonstrating superior performance.

Keywords: ship model; additive manufacturing; scheduling; linear programming; genetic algorithm



Citation: Kim, K.; Kwon, S.; Choi, M. Optimization of Production Scheduling for the Additive Manufacturing of Ship Models Using a Hybrid Method. *J. Mar. Sci. Eng.* **2024**, *12*, 1961. <https://doi.org/10.3390/jmse12111961>

Academic Editors: Zaili Yang, Cristiano Fragassa, Sean Loughney, Eduardo Blanco-Davis and Milad Armin

Received: 26 August 2024

Revised: 17 October 2024

Accepted: 24 October 2024

Published: 1 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Although computational fluid dynamics (CFD) has demonstrated high accuracy in evaluating the hydrodynamic performance of ships, ship model testing remains essential due to physical environmental challenges that are difficult to address with CFD, such as turbulence modeling, grid resolution, and numerical approximation. With the recent emergence of low-emission ships, hull shapes are becoming more diverse to accommodate additional elements for different power systems, such as increased battery sizes. This diversification leads to an increased demand for model tests. However, research on the optimization and efficiency of manufacturing costs related to ship models is insufficient.

Conventional ship model manufacturing methods include the use of fiber-reinforced plastics (FRPs) and wood. FRP, a glass fiber-reinforced plastic, is favored for its high corrosion resistance, lightweight properties, and durability, making it well suited for model construction [1]. However, FRP has significant drawbacks: it is flammable and releases toxic gases upon disposal. Moreover, its production and disposal processes contribute to environmental pollution. The quality of the final product is highly dependent on the skill level of the worker, as the hand layout technique is commonly employed [2].

The technique of using computer numerical control (CNC) machines to cut wood for model fabrication has been a reliable and precise method in the industry for a long time. CNC machines are also sometimes used in the production of FRP models, offering faster production times and reduced labor compared to manual methods since the machine can execute tasks based on the provided input model. However, the high initial costs and space requirements for large cutting equipment can be a barrier for small businesses or individuals [3].

The recent accessibility of additive manufacturing or 3D printing technologies has significantly accelerated its growth [4]. This technique fabricates 3D models by layering and bonding materials such as metals, plastics, powders, and liquids. Unlike traditional casting or machining, which require skilled labor, 3D printing allows anyone to manufacture products. Its widespread use in various industries [5–10] is due to its capacity for freeform design and quick adaptation to design changes, which shortens production times. Choi et al. [3] introduced a modular AM approach for producing ship models. They chose a modular method due to the high initial costs and size limitations of 3D printers, which can hinder the production of certain models. While integrated methods can simplify the process by printing models as a single piece, modularity provides a more practical solution under these constraints.

The modular nature allows ship models to be produced in parallel across multiple machines, significantly reducing the production time through simultaneous manufacturing. However, this approach increases the importance of production scheduling, as scheduling decisions can greatly impact the overall production time. By strategically allocating resources such as 3D printers, materials, and manpower, production supervisors can maximize productivity and throughput while minimizing costs and idle time.

There has been research on scheduling optimization in AM. For example, Chergui et al. [11] addressed the parallel production scheduling problem in AM, and Choi et al. [3] employed a heuristic for scheduling the production of modular ship models, determining the module allocation and making sequencing decisions simultaneously. However, previous studies did not consider the availability of the workers needed to prepare for the next production task upon completion of a module. This factor adds complexity to the optimization of production scheduling for modular ship models.

In this study, we propose an optimization method for the efficient production scheduling of ship models in AM. Our method aims to minimize the production time of the final module by optimizing the module allocation and sequencing while considering worker availability. To achieve this, we employ a hybrid approach that combines linear programming (LP) and a genetic algorithm (GA), selectively using either LP or GA based on the number of modules being produced to leverage the strengths of both techniques. To validate the efficacy of our approach, we compare the proposed method with a heuristic approach through case studies.

2. Scheduling Challenges in Additive Manufacturing with Consideration of Workers' Availability

Traditionally, scheduling was not a primary concern in the monolithic AM process, where entire objects were typically produced as single prints. However, efforts to enhance the efficiency of AM have spurred exploration of various strategies. This includes dividing oversized products exceeding the capacity of AM, or employing multiple machines to increase the production volume [12], leading to the emergence of configurations like 3D printing farms. With the advent of modular AM, where small modules are produced using multiple AM machines and assembled to create larger products, scheduling has become a crucial consideration for optimizing the production process. Scheduling, a significant aspect of traditional manufacturing processes, is increasingly gaining importance in AM as it evolves toward formats resembling 3D printing farms. With the emergence of modular AM production, where small modules are produced using multiple AM machines and then assembled to create larger products, scheduling has become an important consideration for optimizing the production process. Li et al. [13] addressed the production planning problem in AM using a parallel production approach that assigns multiple parts to groups and allocates them to machines. Ransikarbum et al. [14] used a mixed integer linear programming method to optimize the allocation of modules to machines for multipurpose optimization. Dvorak et al. [15] explained and modeled the key optimization problem of minimizing the production time using layered manufacturing, operational research, and

artificial intelligence. A mathematical approach was proposed for scheduling the build and post-processing processes for decomposed parts in layered manufacturing [16].

In addition to these AM-specific studies, broader operational research (OR) techniques offer valuable insights for scheduling in modular AM processes. Our work is similar to the blocking job shop scheduling problem (BJSSP), which addresses scheduling delays caused by resource constraints, particularly when tasks are blocked from proceeding to the next stage due to limited machine capacity. Relevant studies include [17–19]. These studies explore the complexities of blocking effects, which are also present in modular AM when machines become bottlenecks.

Furthermore, recent advancements in hybrid algorithms, such as the hybrid differential evolution (HDE) algorithm [20], have been proposed to handle resource-constrained project scheduling with flexible project structures. The combination of differential evolution and forward–backward improvement demonstrates strong potential for managing complex scheduling problems in resource-limited environments. Similarly, the adaptive large neighborhood search with constraint programming (ALNS-CP) algorithm was proposed for flexible job shop scheduling [21], which provides an efficient approach to address multiple resource constraints in production planning. An improved evolutionary algorithm for parallel batch processing machine scheduling was introduced [22], which combines GAs with heuristic placement strategies to optimize both part allocation and placement in AM systems. Rohaninejad et al. [23] also developed a hybrid learning-based meta-heuristic algorithm for the scheduling of an AM system consisting of parallel SLM machines, combining NSGA-II with k-means clustering and a regression neural network to enhance the scheduling efficiency. These methods can be particularly useful in modular AM, where tasks must be efficiently scheduled across multiple machines with varying capabilities.

Lastly, as the importance of production planning for AM has grown, it has become important to accurately predict the production time for 3D printing. To address this problem, machine learning-based predictions have been developed to facilitate efficient production planning [24,25].

In this paper, the objective of scheduling optimization is to minimize the total production time, known as the makespan, by simultaneously making module assignment and sequencing decisions. Figure 1 illustrates the optimization problem.

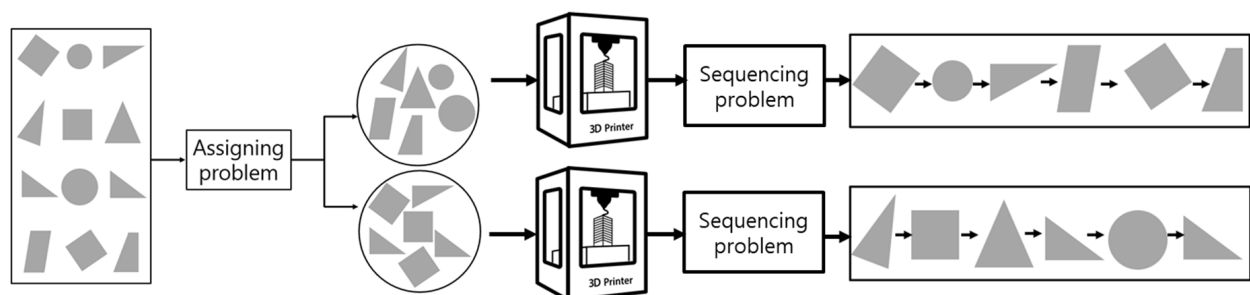


Figure 1. An example of assigning and sequencing problems.

In defining the mathematical model, we make the following assumptions: (1) the required production time for each module is known, (2) multiple identical machines are used, (3) the production setup time is uniform, and (4) each machine can produce only one module at a time. Based on these assumptions, the model is constructed, incorporating flexible production start times and task time limits.

Let us consider an example problem involving two AM machines and four modules, each with different production times, as shown in Table 1. When these modules are assigned to the two machines, various schedules can be created depending on the allocation and sequencing of these modules, as illustrated in Figure 2. The numbers on the bar graph represent each module number and the red dotted line represents the production completion time. As shown in Figure 2a, the Gantt chart illustrates an example where

modules 1 and 3 are produced sequentially on printer 1, while modules 2 and 4 are produced sequentially on printer 2. Similarly, Figure 2b–d depict Gantt charts representing different cases of module assignments and production sequences on the printers. These examples demonstrate how the total production time varies depending on the assignment and production sequence of the modules.

Table 1. Required build time of modules.

Module Index	1	2	3	4
Time (minute)	500	700	800	900

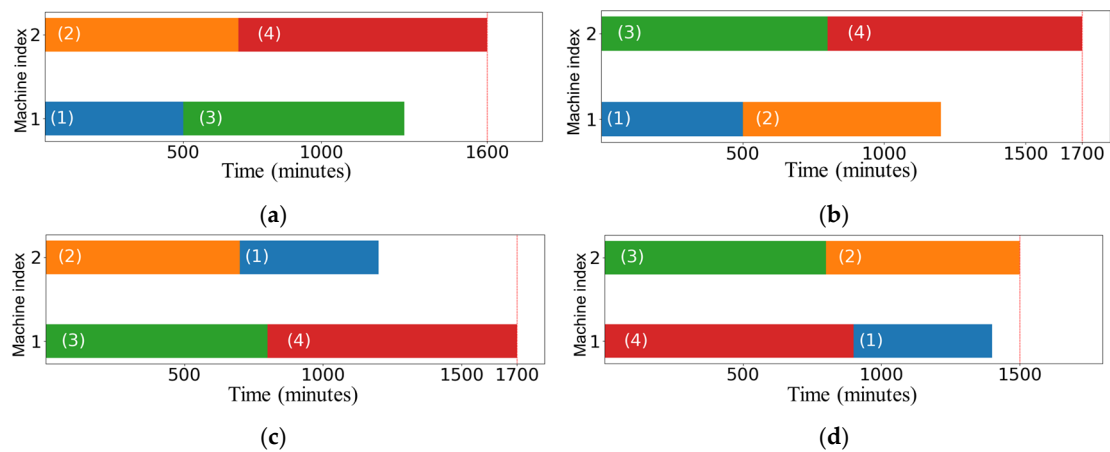


Figure 2. Impact of module allocation and sequencing on makespan variability.

However, the previous examples did not consider workers' available time. Figure 3 shows workers' available time, divided into start and finish times. The production start time for each module must fit within these divisions. Figure 4a–d show the revised schedules adjusted for workers' availability based on the initial schedules shown in Figure 2a–d.

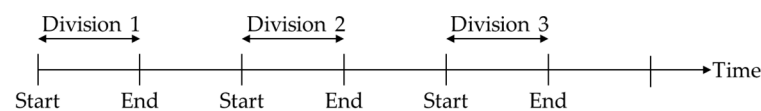


Figure 3. An example of production constraints by time division.

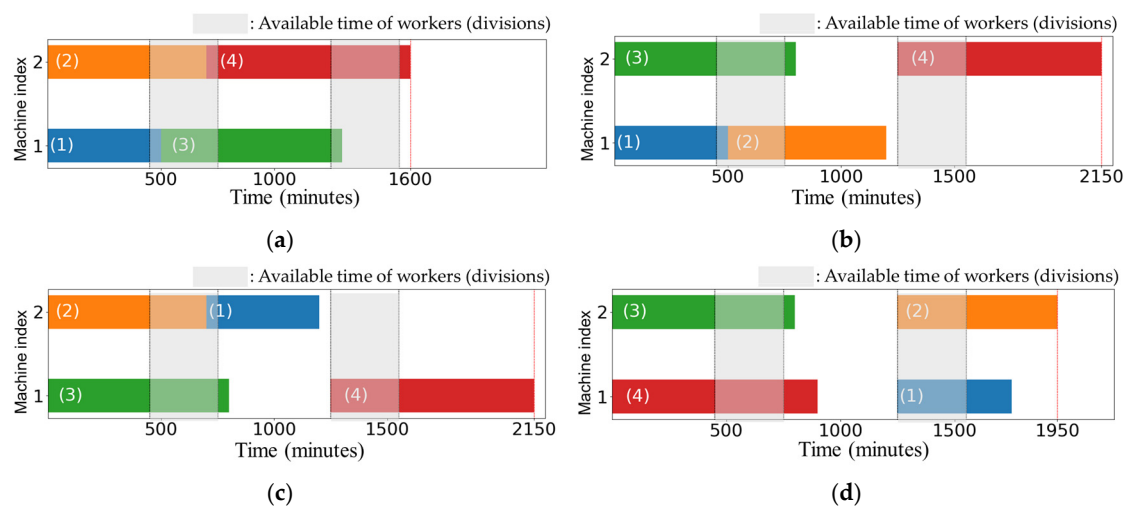


Figure 4. Schedules adjusted for workers' availability.

The adjusted completion times in Figure 4 reveal an interesting result. Despite identical module assignments and sequencing decisions, the completion times differ significantly. While Figure 2d initially shows the shortest production time without considering workers' availability, Figure 4a becomes the shortest when accounting for workers' available time. This demonstrates that the makespan can vary significantly when solutions derived without considering workers' availability are implemented in real-world scenarios.

3. Hybrid Method for Parallel Production Scheduling for Additive Manufacturing

3.1. Mathematic Definition Using Linear Programming

To solve the module assignment and sequencing problem, we propose a network-based model commonly used in scheduling problems, as seen in works like Choi [26]. This model comprises nodes and arcs, as illustrated in Figure 5. In this context, each network corresponds to a machine, with nodes representing modules and arcs indicating the production sequence. The source node signifies the starting point of production, and the sink node represents the endpoint. Other nodes represent modules. Each module must be processed exactly once and cannot revisit the same node. An arc originating from the source node and connecting to a module node indicates the commencement of production for that module. Conversely, an arc leading from a module node to the sink node signifies the completion of production of the machine. Additionally, the sets A_{po} , A_{pi} , A_{dh} are all subsets of set A , in which A_{po} represents the arcs going out from the source node to module nodes, A_{pi} represents the arcs going into the sink node from the module nodes, and A_{dh} represents the arcs between the modules.

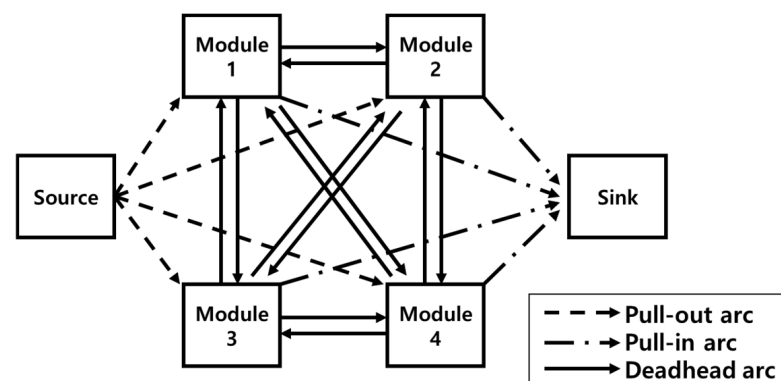


Figure 5. Network model and subset (A_{po}, A_{pi}, A_{dh}) .

We mathematically describe the optimization problem using the following sets, parameters, and variables.

Sets:		
N		Nodes, indexed by i or j
J		Nodes associated with modules, indexed by i or j
R		AM machines, indexed by k
A		Arcs, indexed by (i, j)
A_{po}		Pull-out arcs, indexed by (i, j)
A_{pi}		Pull-in arcs, indexed by (i, j)
A_{dh}		Deadhead arcs, indexed by (i, j)
D		Divisions, indexed by d
Parameters:		
T_i^{BUILD}		The production time of module i
T_d^{ST}		Start time of division d
T_d^{FN}		Finish time of division d
M		The big number
Variables:		

x_{ijk}	1 if arc (i, j) of AM machine k is selected, 0 otherwise
y_{id}	1 if division d selected by module i , 0 otherwise
u_i	Production start time of module i
z	The latest completion time among the AM machines
Model:	
Minimize	z (1)
s.t. $\sum_{k \in R} \sum_{(i,j) \in A_{po} \cup A_{dh}} x_{ijk} = 1$	$j \in J$ (2)
$\sum_{(i,j) \in A_{po}} x_{ijk} = \sum_{(i,j) \in A_{pi}} x_{ijk}$	$k \in R$ (3)
$\sum_{(i,j) \in A_{po}} x_{ijk} \leq 1$	$k \in R$ (4)
$\sum_{(i,j) \in A_{po} \cup A_{dh}} x_{ijk} = \sum_{(i,j) \in A_{pi} \cup A_{dh}} x_{ijk}$	$j \in J, k \in R$ (5)
$\sum_{(i,j) \in A_{dh}} x_{ijk} \leq M \sum_{(i,j) \in A_{po}} x_{ijk}$	$k \in R$ (6)
$u_i + T_i^{BUILD} \leq z$	$i \in J$ (7)
$u_i + T_i^{BUILD} \leq u_j + M(1 - x_{ijk})$	$(i, j) \in A, k \in R$ (8)
$\sum_{d \in D} y_{id} = 1$	$i \in J$ (9)
$\sum_{d \in D} T_d^{ST} y_{id} \leq u_i \leq \sum_{d \in D} T_d^{FN} y_{id}$	$i \in J$ (10)
$x_{ijk} \in \{0, 1\}$	$(i, j) \in A, k \in R$ (11)
$0 \leq u_i$	$i \in N$ (12)

Equation (1) is the objective function of this model, which aims to minimize the module z that is produced the latest. Equations (2)–(12) represent the production constraints of multiple AM machines using a network model. Equation (2) indicates that each module must be produced once, ensuring that all the modules are produced. Equations (3)–(6) represent network flow conservation constraints. Equation (3) states that the number of arcs entering the sink node should be equal to the number of arcs leaving the source node. Equation (4) represents a constraint in which each machine should have only one arc from the source node to a different node. Equation (5) indicates that the number of arcs assigned to the source and module nodes is equal to the number of arcs assigned to the sink module node. Equation (6) is a constraint that mandates the existence of an arc leading to the sink node if there is an arc between the modules in each machine. Equation (7) sets the start time of each module as u_i and defines z as the sum of the production time and the module that takes the latest time to produce. Equation (8) is a constraint that requires the production start time of module j to be after the production start time of the previous module i . Constraints (9) and (10) are defined to limit the working times. In this model, the concept of segments has been introduced. The available and unavailable working times are also segmented, and the production start time of each module must be within the feasible time division. Equation (9) is a constraint that requires each module to belong to one of the divisions, and Equation (10) is a constraint that requires the production start time of each module to be within the specified division range. Equations (11) and (12) represent constraints on the range of variables.

3.2. Hybrid Method to Overcome the Size Limitations of Linear Programming

LP offers the advantage of mathematically proving the optimality of solutions and providing quick results, making it a powerful tool for optimization problems. However, the optimization time for LP increases exponentially as the problem size grows. To address this limitation, we adopted a GA as an alternative method for scenarios where the number of modules exceeds what LP can handle in a reasonable time. GAs are well known for their effectiveness in finding solutions for complex problems with various constraints due to their adaptability. However, despite their global search capabilities, GAs often require relatively long processing times for simpler problems and can become stuck in local optima.

To leverage the strengths of both LP and GAs, we employ a hybrid method. The LP is used for problems when the problem size is below a certain threshold, and the GA is used when the problem size exceeds this threshold. We conducted experiments to determine the appropriate threshold number, investigating the computation times as the problem size increased. As shown in Figure 6a, the experimental results indicate that when the number of modules is 14, the computation is completed in approximately 1 s. However, when the number of modules increases to 15, the computation time sharply rises to around 1500 s. This significant increase is due to the rapid growth in computational complexity as the number of modules increases. Therefore, we set 15 modules as the threshold value. Based on this threshold, LP is applied to problems with fewer than 15 modules, while the GA optimization method, as illustrated in Figure 6b, is employed for problems with 15 or more modules.

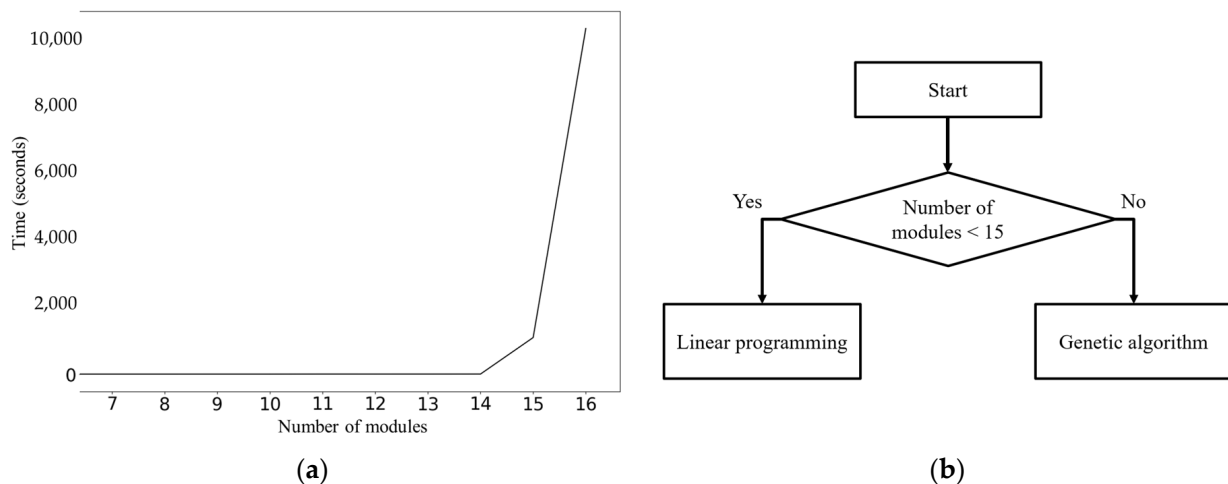


Figure 6. (a) Experiments on the computation time according to the number of modules and (b) the method selection algorithm for the hybrid method.

It is crucial to represent the solutions effectively, which is determined by the chromosome structure in the GA. Figure 7 illustrates the chromosome structure used in the GA. Each gene represents a module, containing information about the machine assigned to build the module and its production priority on that machine. The priority is a value between 0 and 1.

The chromosome representation scheme shown in Figure 7 illustrates how each gene corresponds to a specific module. Each gene contains information about the machine to which the module is assigned and the production priority of that module on the machine. After grouping the modules by their assigned machine, the priority values of the modules are compared within each group. Modules with lower priority values are produced first. This process is repeated across all the machines to determine the final allocation and production order. During the initial population generation, gene values are assigned randomly, which may result in imbalances, such as some machines being overloaded with modules while others have none. These initial schedules are then adjusted to consider

workers' available time for starting production tasks. Figures 7 and 8 illustrate this schedule conversion. Figure 8a shows the allocation of modules to machines and their arrangement by priority without considering worker's available time. Figure 8b illustrates the adjusted schedule that accounts for worker's available time.

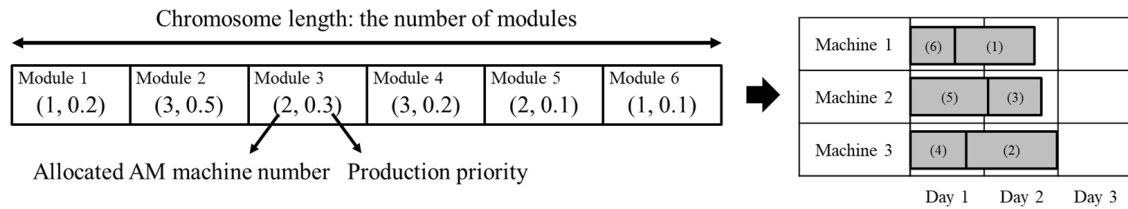


Figure 7. Chromosome representation scheme.

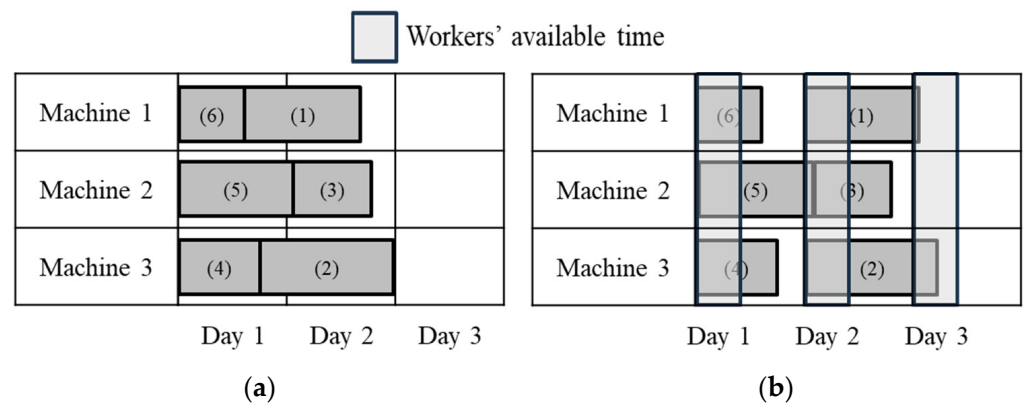


Figure 8. The adjusted schedule that considers workers' available time: the numbers in the grey boxes indicate assigned module numbers.

The GA employs a one-point crossover strategy, where the cutting point can occur anywhere along the chromosome. Figure 9 illustrates an example of one-point crossover, with the cutting point located in the middle of the parent chromosomes.

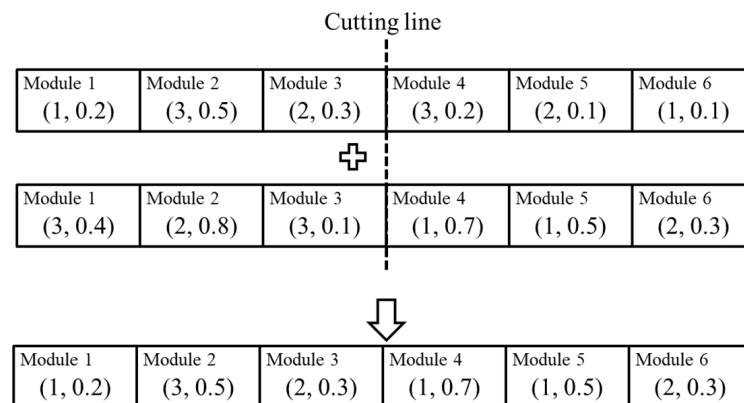


Figure 9. An example of a one-point crossover strategy.

During mutation, the GA randomly determines the number of genes to be altered and selects those genes accordingly. The selected genes are then replaced with randomly generated genes. This process helps in achieving global optimization. The overall flow of the GA is depicted in Figure 10.

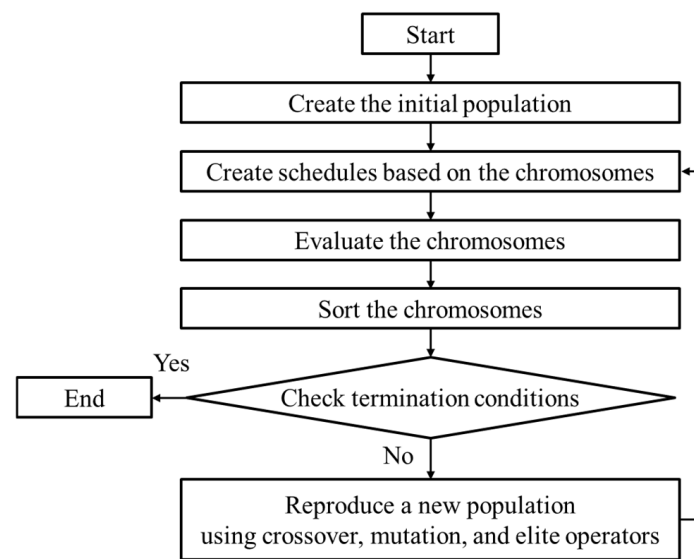


Figure 10. The overall process of the GA.

4. Case Study

In the case study, we selected the ship model from the previous study by Choi et al. [3] and applied our proposed method to its production. The ship's dimensions, including the length overall (L.O.A), length between perpendiculars (L.B.P), breadth (B), depth (D), and draft (T), are provided in Table 2.

Table 2. Detailed specifications of the ship model [3].

Subject	Ship	Model (Scale Ratio, $\lambda = 10$)
L.O.A [m]	17.6	1.760
L.B.P [m]	12.32	1.232
B [m]	3.98	0.398
D [m]	0.95	0.095
T [m]	0.66	0.066

The authors removed the ship's interior and divided the ship into 32 modules to fit six identical AM machines with a maximum size of $300 \times 250 \times 250$ mm, as shown in Figure 11.

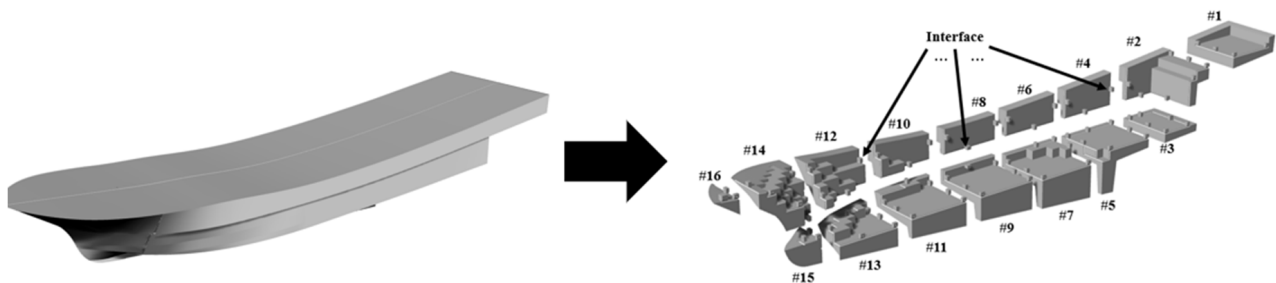


Figure 11. Modularized ship model.

The production time for each module is specified in Table 3. To account for potential uncertainties in production, such as variations in the operating conditions and machine performance, an additional 5% buffer time allowance has been included for each module. Each module must be assigned to one of the six AM machines for production, and each module is produced once. [3] used a heuristic method for the production scheduling problem of the modular ship model. The heuristic method assigned the modules using

the module index to the available machines in order. Using this heuristic approach, the makespan was 9813 min.

Table 3. Production time required for each module.

Module Index	Time (min)	Module Index	Time (min)
1, 17	1483	9, 25	1534
2, 18	1711	10, 26	1179
3, 19	1038	11, 27	1408
4, 20	1064	12, 28	1922
5, 21	1510	13, 29	2209
6, 22	873	14, 30	2613
7, 23	1575	15, 31	756
8, 24	745	16, 32	756

To validate the performance of the proposed method, we applied it to the same scheduling problem. Given that the ship model comprises 32 modules, exceeding the threshold of 15, the hybrid method utilized the GA. The optimization algorithms were executed on a CPU with an Intel(R) Core(TM) i7-13700HX processor operating at 3.70 GHz and 16.0 GB of memory. The GA hyperparameters were configured as follows: the population size was 50, with 7 elite individuals retained from the top-performing solutions of the previous generation, 27 individuals generated through crossover, and 16 individuals generated through mutation. The termination criterion for the GA was set to halt reproduction after generating 3000 solutions. The convergence curve for each generation is shown in Figure 12.

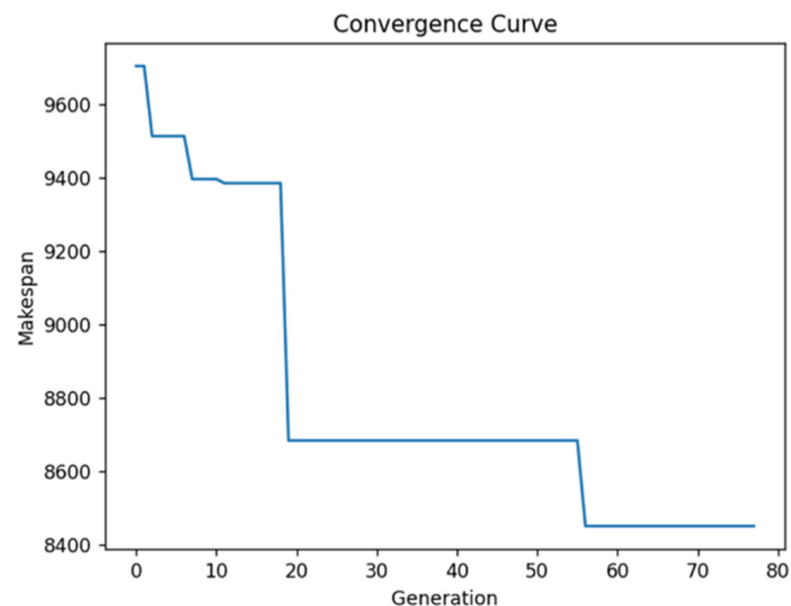
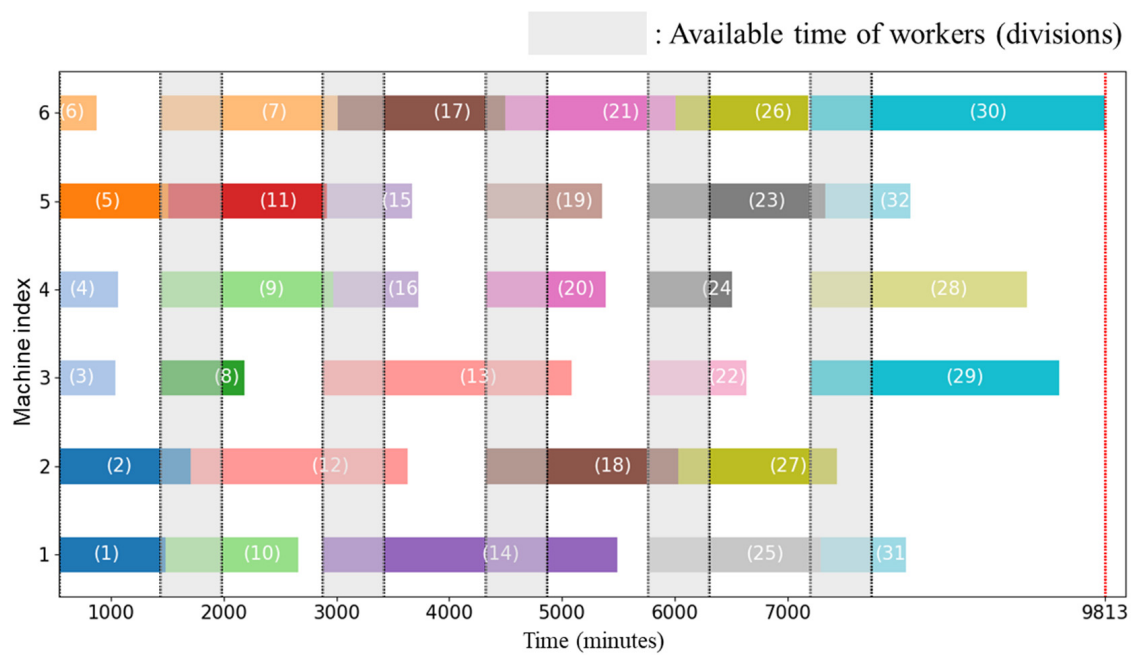
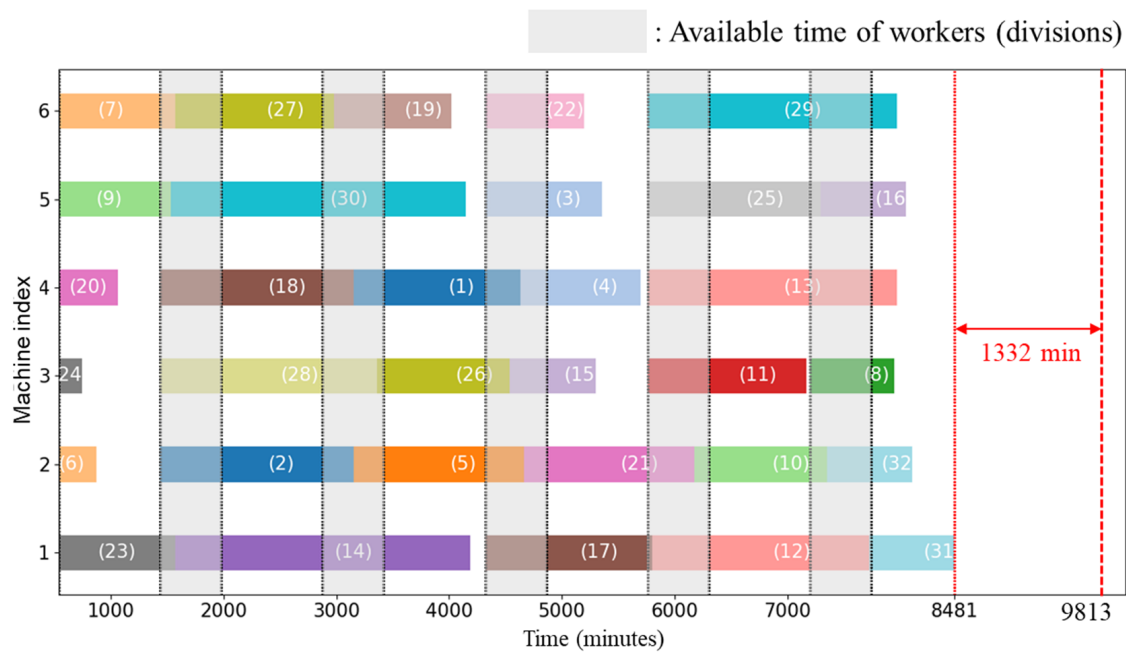


Figure 12. Convergence curve of the GA.

Due to the stochastic nature of the GA, the results can vary with each run. Empirically, repeating the GA 10 times yielded the best solution. Each iteration took an average of 27 s, depending on the number of modules. This demonstrated that the scheduling results could be obtained in a sufficiently short time. Additionally, by running the GA multiple times, the best, worst, and average results were recorded. The makespan through the GA was 8481 min for the best result, 9396 min for the worst result, and 8974.1 min on average. Compared to the heuristic approach, this shows a maximum difference of 1332 min, or approximately 13.58%. The comparison of the scheduling results is described alongside a Gantt chart in Figure 13.



(a) The optimal schedule determined by the heuristic method.



(b) The optimal schedule determined by the hybrid method.

Figure 13. Scheduling results represented by Gantt charts.

We also applied the proposed method to a different problem, where the same ship model was divided into larger modules, assuming different numbers of machines and build capabilities. Under these new assumptions, the ship model was divided into 12 modules to be assigned to three machines. Table 4 presents the production times for each module.

Table 4. Module production times.

Module Index	1	2	3	4	5	6	7	8	9	10	11	12
Time (minutes)	1483	1711	1711	1038	1064	1064	1510	1510	873	1575	1575	745

For this problem, the hybrid method used LP instead of the GA because the number of modules was below the threshold. The optimal schedule derived by the hybrid method is presented in Figure 14. Using LP ensured the optimality of the solution, and the optimization process took only 0.2 s, highlighting the efficiency of LP for problem sizes under the threshold for AM processes.

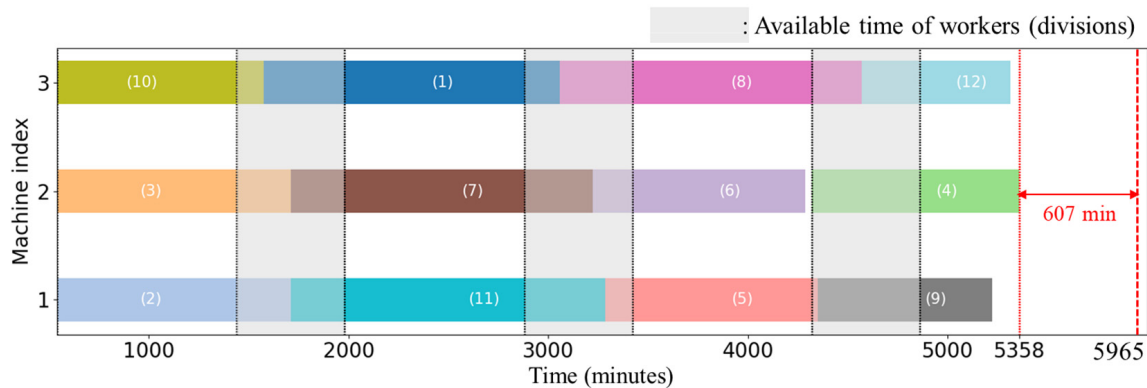
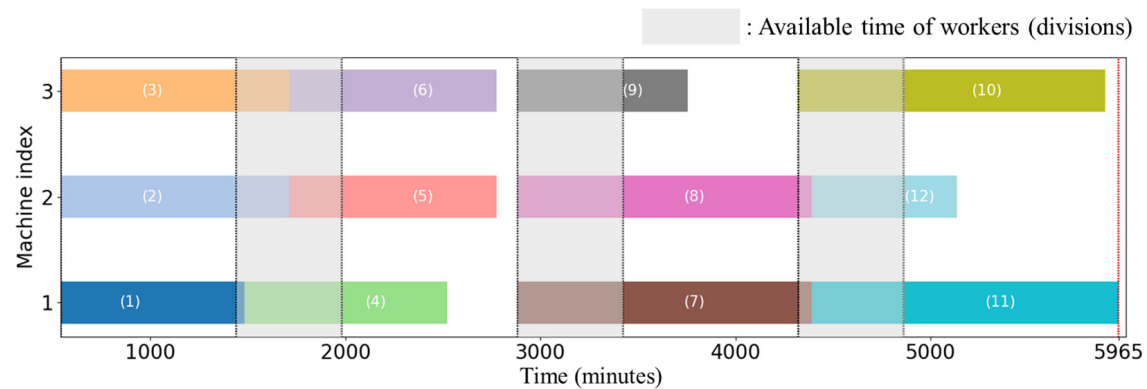


Figure 14. Scheduling results represented by Gantt charts for a reduced number of modules.

In comparison, the makespan derived by the heuristic method was 5965 min, whereas the proposed method yielded a makespan of 5358 min. This reduction of 607 min (approximately 10.02%) demonstrates the effectiveness of the proposed method in optimizing the production schedule, both in terms of the computation time and the solution quality.

5. Conclusions

This paper presented a hybrid method that employs either linear programming (LP) or a genetic algorithm (GA) to tackle the parallel production scheduling problem in additive manufacturing (AM) for modularized ship models, taking into account workers' available time. While some may view the scheduling of ship model production as non-critical, ineffective scheduling can lead to significant wasted time. For manufacturers operating 3D printing farms and producing multiple ship models for various customers, the proposed method is particularly powerful because it can determine the optimal schedule under conditions involving numerous printers and modules. This research provides a solid foundation for further exploration and development.

The case study demonstrated the proposed method's effectiveness by applying it to a ship model from earlier research [3]. The new method significantly reduced the makespan when compared to the heuristic approach. In a scenario involving 32 modules, the GA-derived schedule achieved the best makespan of 8481 min, a substantial improvement over the heuristic method's 9813 min. Furthermore, when the ship model was divided into

12 larger modules and assigned to three machines, the hybrid method using LP resulted in a makespan of 5358 min, compared to 5965 min with the heuristic method. This reduction of 607 min (approximately 10.02%) underscores the method's ability to optimize production schedules effectively, both in terms of the computation time and the solution quality.

However, there is room for improvement. In this paper, we assumed fixed production times and no failures. In practice, the estimated production time of modules can be uncertain, and this uncertainty can vary depending on the operating software for the 3D printers. To determine more robust solutions, the model needs to account for uncertainties caused by estimation errors and potential production failures. In particular, addressing the variability in production times is a crucial area for future research, as real-world manufacturing conditions often involve fluctuations due to machine performance and operational factors. Incorporating this variability would significantly enhance the applicability and robustness of the proposed scheduling method.

Moreover, this study did not consider potential scheduling conflicts due to pre-existing tasks or maintenance schedules. In industrial settings, machines are often shared between multiple tasks or undergo maintenance, leading to scheduling challenges. Future research should incorporate these factors to further improve the real-world relevance and applicability of the proposed method.

Additionally, we assumed the homogeneous build capabilities of 3D printers and no production failures or pre-existing tasks. However, in real-world scenarios, companies often operate a variety of 3D printers with different capacities and may encounter unexpected failures or have ongoing commitments that need to be managed. This diversity in machine capabilities, along with the possibility of failures and varying availability, must be considered to create a truly effective scheduling method. Future research will address these factors to improve the robustness and real-world applicability of the proposed model. While the proposed method has shown potential in controlled experimental settings, further validation in real-world manufacturing environments is necessary. Testing the model in diverse industrial settings will help assess its robustness and applicability under practical conditions, where machine variability, production failures, and operational constraints are prevalent. Such testing will ensure that the model can deliver optimized scheduling solutions in real-world scenarios.

Lastly, an optimized solution can be achieved when the scheduling problem is handled together with the module division problem. The scheduling results are highly dependent on the module division decisions. Therefore, future research should integrate these two consecutive problems into a single, unified problem to improve the overall efficiency and effectiveness.

Author Contributions: K.K.: conceptualization, methodology, data curation, formal analysis, methodology, investigation, resources, software, validation, visualization, writing—original draft, writing—review and editing. M.C.: conceptualization, methodology, formal analysis, visualization, supervision, validation, project administration, writing—review and editing, funding acquisition. S.K.: conceptualization, formal analysis, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work received partial support from the Ministry of Trade, Industry & Energy of Korea. Firstly, through the HRD program for industrial innovation, specifically the 'Smart Yard Professional Human Resources Training Project' (P0017006), and secondly, via the project (20018667) focusing on 'Development of an Integrated Ship Design System for Hull, Compartment, Basic Calculation, and Loading Guidance Using Artificial Intelligence Technology'.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Sathishkumar, T.; Satheeshkumar, S.; Naveen, J. Glass fiber-reinforced polymer composites—A review. *J. Reinf. Plast. Compos.* **2014**, *33*, 1258–1275. [\[CrossRef\]](#)
2. Coackley, N. *Fishing Boat Construction, 2: Building a Fiberglass Fishing Boat*; No. 507; Food & Agriculture Org: Rome, Italy, 1991.
3. Choi, L.; Choi, M.; Kwon, S.; Youn, D.; Song, G. Modular production of small ship models using 3D printing for model tests. *Ocean Eng.* **2024**, *302*, 117685. [\[CrossRef\]](#)
4. Tofail, S.A.M.; Koumoulos, E.P.; Bandyopadhyay, A.; Bose, S.; O'Donoghue, L.; Charitidis, C. Additive manufacturing: Scientific and technological challenges, market uptake and opportunities. *Mater. Today* **2018**, *21*, 22–37. [\[CrossRef\]](#)
5. Ventola, C.L. Medical applications for 3D printing: Current and projected uses. *Pharm. Ther.* **2014**, *39*, 704–711.
6. Sun, J.; Zhou, W.; Huang, D.; Fuh, J.Y.H.; Hong, G.S. An Overview of 3D Printing Technologies for Food Fabrication. *Food Bioprocess Technol.* **2015**, *8*, 1605–1615. [\[CrossRef\]](#)
7. Joshi, S.C.; Sheikh, A.A. 3D printing in aerospace and its long-term sustainability. *Virtual Phys. Prototyp.* **2015**, *10*, 175–185. [\[CrossRef\]](#)
8. Attaran, M. The rise of 3-D printing: The advantages of additive manufacturing over traditional manufacturing. *Bus. Horiz.* **2017**, *60*, 677–688. [\[CrossRef\]](#)
9. Ziółkowski, M.; Dyl, T. Possible applications of additive manufacturing technologies in shipbuilding: A review. *Machines* **2020**, *8*, 84. [\[CrossRef\]](#)
10. Taşdemir, A.; Nohut, S. An overview of wire arc additive manufacturing (WAAM) in shipbuilding industry. *Ships Offshore Struct.* **2021**, *16*, 797–814. [\[CrossRef\]](#)
11. Chergui, A.; Hadj-Hamou, K.; Vignat, F. Production scheduling and nesting in additive manufacturing. *Comput. Ind. Eng.* **2018**, *126*, 292–301. [\[CrossRef\]](#)
12. Luo, L.; Baran, I.; Rusinkiewicz, S.; Matusik, W. Chopper. *ACM Trans. Graph.* **2012**, *31*, 129. [\[CrossRef\]](#)
13. Li, Q.; Kucukoc, I.; Zhang, D.Z. Production planning in additive manufacturing and 3D printing. *Comput. Oper. Res.* **2017**, *83*, 157–172. [\[CrossRef\]](#)
14. Ransikarbum, K.; Ha, S.; Ma, J.; Kim, N. Multi-objective optimization analysis for part-to-Printer assignment in a network of 3D fused deposition modeling. *J. Manuf. Syst.* **2017**, *43*, 35–46. [\[CrossRef\]](#)
15. Dvorak, F.; Micali, M.; Mathieug, M. Planning and scheduling in additive manufacturing. *Intel. Artif.* **2018**, *21*, 40–52. [\[CrossRef\]](#)
16. Oh, Y.; Cho, Y. Scheduling of build and post processes for decomposed parts in additive manufacturing. *Addit. Manuf.* **2022**, *59*, 103164. [\[CrossRef\]](#)
17. Dabab, A.; Bendjoudi, A.; AitZai, A.; Taboudjemat, N.N. Efficient Parallel Tabu Search for the Blocking Job Shop Scheduling Problem. *Soft Comput.* **2019**, *23*, 13283–13295. [\[CrossRef\]](#)
18. Mogali, J.K.; Barbulescu, L.; Smith, S.F. Efficient Primal Heuristic Updates for the Blocking Job Shop Problem. *Eur. J. Oper. Res.* **2021**, *295*, 82–101. [\[CrossRef\]](#)
19. Oddi, A.; Rasconi, R.; Cesta, A.; Smith, S. Iterative Improvement Algorithms for the Blocking Job Shop. In Proceedings of the International Conference on Automated Planning and Scheduling, Atibaia, Sao Paulo, Brazil, 25–29 June 2012; Volume 22, pp. 199–206.
20. van der Beek, T.; Souravlias, D.; van Essen, J.; Pruyn, J.; Aardal, K. Hybrid differential evolution algorithm for the resource constrained project scheduling problem with a flexible project structure and consumption and production of resources. *Eur. J. Oper. Res.* **2024**, *313*, 92–111. [\[CrossRef\]](#)
21. Kasapidis, G.A.; Paraskevopoulos, D.C.; Mourtos, I.; Repoussis, P.P. A unified solution framework for flexible job shop scheduling problems with multiple resource constraints. *Eur. J. Oper. Res.* **2024**, *320*, 479–495. [\[CrossRef\]](#)
22. Zhang, J.; Yao, X.; Li, Y. Improved evolutionary algorithm for parallel batch processing machine scheduling in additive manufacturing. *Int. J. Prod. Res.* **2020**, *58*, 2263–2282. [\[CrossRef\]](#)
23. Rohaninejad, M.; Tavakkoli-Moghaddam, R.; Vahedi-Nouri, B.; Hanzálek, Z.; Shirazian, S. A hybrid learning-based meta-heuristic algorithm for scheduling of an additive manufacturing system consisting of parallel SLM machines. *Int. J. Prod. Res.* **2022**, *60*, 6205–6225. [\[CrossRef\]](#)
24. Di Angelo, L.; Di Stefano, P. A neural network-based build time estimator for layer manufactured objects. *Int. J. Adv. Manuf. Technol.* **2011**, *57*, 215–224. [\[CrossRef\]](#)
25. Oh, Y.; Sharp, M.; Sprock, T.; Kwon, S. Neural network-based build time estimation for additive manufacturing: A performance comparison. *J. Comput. Des. Eng.* **2021**, *8*, 1243–1256. [\[CrossRef\]](#)
26. Choi, M. A hybrid method for the determination of the minimum number of transport vehicles in a shipyard. *J. Korean Soc. Mar. Eng.* **2021**, *45*, 51–59. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.